

Lenguaje de Programación: Recursión

José Luis Alonzo Velázquez

Universidad de Guanajuato

Sesión 13

Recursión en Karel

La recursión, al igual que los ciclos, nos sirve para que Karel ejecute un conjunto de instrucciones repetidamente. La diferencia con los ciclos es que la recursión está definida únicamente para funciones.

La sintaxis es:

```
has_algo()  
  {  
    <instrucción>  
    <instrucción>  
    ⋮  
    has_algo();  
    <instrucción>  
    ⋮  
    <instrucción>  
  }
```

Nota:

A primera vista puede parecer extraño y hasta ridículo. Pero esta es una de las herramientas más poderosas en programación.

Recursión en el mundo real.

Un ejemplo, es cuando ponemos dos espejos que están uno frente al otro, el primero de ellos refleja lo que se ve en el segundo, pero el segundo espejo refleja lo que se ve en el primero. Si nos colocamos en medio de los dos espejos, podremos ver muchas imágenes de nosotros cada una más pequeña y lejana que la otra.



Figura: Espejo frente a otro espejo.

Recursión en el mundo de Karel.

Supongamos que queremos que Karel recoja un beeper que se encuentra en su misma calle, pero no sabemos a que distancia esta. Entonces podemos hacer una función que se llame **busca_beeper**. El caso más sencillo es que Karel ya este sobre el beeper en cuyo caso ni debería hacer nada.

Código para hacer la búsqueda mediante recursión

```
define busca_beeper()  
  {  
    if(!nextToABeeper)  
      {  
        move();  
        busca_beeper();  
      }  
  }
```

Equivalencia entre recursión y while

Ahora supongamos que queremos que Karel termine viendo al Norte sin importar a que lado empezó viendo.

La función que ya conocíamos era:

```
define vista_al_norte()  
  {  
    while(notFacingNorth)  
      {  
        turnleft();  
      }  
  }
```

Ahora la versión recursiva sería

```
define vista_al_norte()  
  {  
    if(notFacingNorth)  
      {  
        turnleft();  
        vista_al_norte();  
      }  
  }
```

Nota

La diferencia entre estas dos funciones es muy sutil, aunque ambas llevan a cabo la tarea correctamente. La primera se llama una sola vez y nunca dejamos de ejecutarla hasta que hemos llegado al norte. La segunda crea copias de si misma cada vez que Karel no ve al norte.

Propiedades de las funciones recursivas



Edgar Alfredo Duéñez Guzmán & Edgar Said Hernández
Sánchez & Marte Alejandro Ramírez Ortegón *Los Dilemas de
Karel*. CIMAT, Mayo 2006.